Propositional Stability ensures that when a proposition is transacted between two logics (more on this later) - it never acquires a new truth-value beyond those it could have already acquired under the first logic under which it is evaluated   
  
where ⊶ ∈ ℕ   
where ⋇ ∈ {a, ..., z, ...} | {a, ..., z, ...} = ℕ   
  
**Conventions**  
  
We write ML⊶⋇ to denote a semantics (model or truth-assignment M) for a language L⊶ with ⋇-many truth values.   
  
We write VML⊶⋇(p) to denote a truth-evaluation of p under semantics (model or truth-assignment M) for a language L⊶ with ⋇-many truth values.   
  
We write VML1aVML2b(p)\* to denote any possible truth-evaluation of p to a truth-value t in semantics ML2b such that: t ∈ ML2b and t ∉ ML1a.   
  
An instruction set is a finite procedure or algorithm mapping one input to one output.   
  
**Elaborated Definition**   
  
Propositional stability: a proposition or sentence p evaluated under semantics ML1a will preserve its truth-value under semantics ML2b whenever a ⊆ b and no instruction set exists to map VML1a(p) to any VML1aVML2b(p)\*.   
  
**Initial Result**   
  
Any proposition under a Boolean logic will exhibit propositional stability against a (standard - thus far axiomatized) Kleene 3-Value Algebra. **Proof:** Obvious. No single proposition already assigned a truth-value of 'true' or 'false' can receive a truth-value of 'indeterminate' or 'true and false'. ∎

Originally Posted at: <http://www.postlib.com/propositional-stability/>